IN-18-TM
91479
P.13

(NASA-TM-107899) A STUDY OF KNOWLEDGE-BASED SYSTEMS FOR THE SPACE STATION Report, Oct. 1987 - Jan. 1988 (NASA) 13 p

N92-70676

Unclas
Z9/18 0091479

A Study of Knowledge-Based Systems for the Space Station

PETER FRIEDLAND

AI RESEARCH BRANCH, MAIL STOP 244-17
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035

NASA Ames Research Center
Artificial Intelligence Research Branch

Technical Report RIA-89-01-01-03

January, 1989

A Study of Knowledge-Based Systems for the Space Station

Peter Friedland, NASA Ames Research Center, Moffett Field, California

Gregg Swietek, NASA Headquarters, Washington, D.C.

Bruce Bullock, ISX Corp., Thousand Oaks, California

Abstract

A rapid turnaround study on the potential uses of knowledge-based systems for Space Station Freedom was conducted from October 1987 through January 1988. Participants included both NASA personnel and experienced industrial knowledge engineers. Major results of the study included five recommended systems for the Baseline Configuration of the Space Station, an analysis of sensor hooks and scars, and a proposed plan for evolutionary growth of knowledge-based systems on the Space Station.

1. Overview

1.1 Background

From October 1987 through January 1988 a study was undertaken for the Space Station Level I Strategic Plans and Programs Division. The purpose of this study was to enlist the services of a small group of the country's most experienced practitioners of knowledge-based systems (often called expert systems) to perform a rapid analysis of the current and future potential of such systems on Space Station. Specifically, the group was chartered to make three forms of recommendations:

- A short list of Baseline candidates feasible with current technology. In addition, provide estimates on cost for each such system and a detailed performance plan for building each system.
- An analysis of knowledge-based system evolution on Space Station, including descriptions of recommended hooks and scars to be provided at Baseline.
- Recommendations to the Strategic Plans and Programs Division as to high-leverage ways to use its funding to make possible the evolutionary addition of increasingly more sophisticated knowledge-based systems to Space Station.

The group was chaired by Dr. Peter Friedland, Chief of the Artificial Intelligence Research Branch of Ames Research Center. It included Dr. Jaime Carbonell of Carnegie Group, Dr. David Mishelevich of IntelliCorp, Dr. Bruce Bullock of ISX, Mr. Bradley Allen of Inference, Dr. Robert Engelmore of Stanford University, and Dr. Ben Wah of the University of Illinois. Mr. Gregg Swietek of Space Station Strategic Plans and Programs Division served as Executive Secretary and Task Manager. Each member of the group had at least 10 years experience in knowledge-based systems and had been involved in the design and construction of over a dozen of such systems. In addition, each member of the group had participated in many such rapid turnaround analyses.

1.2 Approach

Our approach consisted of individual study from an extensive reading list of Space Station documents (relating to system and subsystem architecture as well as prior work on automation), followed by an intensive visit to Johnson Space Center and Marshall Space Flight Center for detailed discussions with NASA personnel responsible for all major subsystems and modules. This led to group and individual analysis of specific candidate knowledge-based systems as well as evolutionary hooks (paths for software evolution) and scars (paths for hardware evolution).

The group attempted to make maximum use of previous study results. These provided an encyclopedic listing of hundreds of potential applications of knowledge-based systems technology to Baseline and Evolutionary Space Station. They served as a good starting point by providing a classification of types of functions for such systems on the Space Station. A primary goal of this study was to use the experience of the group to narrow down the enormous list of candidates to a small, manageable set of recommendations for Baseline Space Station. The study also provided plans for how the systems could actually be built within the technical, organization, and cost context of the Space Station Freedom program, based on hard evidence from similar industrial systems.

Finally, a significant assumption of this work was that maximal use of knowledge-based systems to enhance safety, reliability, and productivity, and to minimize life-cycle costs, was a desirable goal.

1.3 Evaluation Criteria

At the highest level, the group believes that three major types of criteria are important in recommending the use of knowledge-based systems:

1. Is there a problem worth solving and will its solution be visibly evident? Clearly, one should not invest significant funds for a knowledge-based system if the problem is a trivial one, or perhaps of only academic interest to any major user group.

2. Can knowledge-based systems technology provide a substantial functional edge where conventional computational technology will not suffice? AI is not needed if conventional control methods or quantitative, algorithmic software does a perfectly adequate job.

3. Can the particular problem be solved under the time, resource, and organizational constraints within which it exists?

1.4 General Observations

Several observations struck us as particularly important during the course of our study:

1. We were extremely pleased (and somewhat surprised) by the overwhelmingly positive attitude of crew toward the use of knowledge-based systems on Space Station. We spent considerable time with 6 crew members of varying backgrounds, although all had flown on Shuttle missions. They expressed a very strong desire to spend as much time as possible on Space Station carrying out scientific experiments and as little time as possible worrying about maintenance and other housekeeping chores. As long as the developers of knowledge-based systems involve crew from the initial prototyping stage on, and systems start in advisory mode and slowly evolve to an in-the-loop control mode, crew will be supportive.

2. We applaud the existing in-house work at NASA in potential applications of knowledge-based systems. NASA has done an excellent job of coming up the learning curve; knowledge-based systems are no longer a new technology to NASA.

3. Much has been said and written about hooks and scars for evolutionary use of knowledge-based systems on Space Station, but details at the level where they can be useful to the Work Package contractors were almost nonexistent.

4. As expected by our group, the only real resistance we observed to the immediate use of current technology knowledge-based systems came from the "institutional" MIS community at NASA. We also noted a desire for standardization of on-board hardware and software rather than taking advantage of the concept of distributed, diverse systems on a fast network. While motivated by the necessity for conservative, solid designs, we do not feel it reflects the maturity of the more distributed and flexible approach.

1.5 General Recommendations to the Space Station Program

Before discussing specific recommendations for Baseline and beyond, we offer several more general suggestions to the Space Station Program:

1. Capitalize on existing expertise within the agency, especially in the form of existing significant projects.

2. Remember that the development environment for knowledge-based systems need not be the same as the delivery environment. There are many successful commercial examples of systems being rapidly developed in specialized languages on specialized hardware and then ported to operational software and hardware. The specialized prototyping tools have their place in the development lifecycle.

3. Keep in mind eventual system reusability during design. For example, a knowledge-based system initially used for FDIR may eventually be used for training novices about the system.

4. Make sure users (crew, ground controllers, etc.) are involved from the very beginning of initial system design. This is important for at least three reasons: speed of system construction, functional ability of the systems, and eventual user trust and acceptance.

5. Begin all systems as advisory to crew and ground-control personnel, but plan for an eventual move to in-the-loop control systems.

6. Fire-wall all applications from harm, but make sure they are integrated into actual systems for eventual transition from advisory to control systems.

7. Facilitate early test-bed interactions with both the Data Management System (DMS) and the Operations Management System (OMS); problems of real-time data and control flow are best dealt with early in the life cycle of knowledge-based systems.

8. Concentrate on applications with potential horizontality to many similar uses, but make sure the first application is built in sufficient depth to have internal champions.

9. Quickly develop a reasonable standard for verification and validation of Baseline knowledge-based systems even if it is identical to current traditional software standards.

10. Make sure to incorporate existing non-AI techniques and algorithms into knowledge-based systems when appropriate.

11. Maintain an external expert group to monitor and facilitate rapid development and acceptance of chosen knowledge-based systems applications. Space Station needs a group not tied to a particular Center, contractor, or other constituency whose only goal is rapid, but sensible use of the technology.

2. Recommendations for Baseline Space Station Systems

During our analysis of prior work and our on-site investigations we gave serious consideration to the potential of most of the dozens of systems previously enumerated. We quickly narrowed the list down to approximately 20 serious possibilities; after much discussion, we decided to limit our specific recommendations for the baseline Space Station to five distinct knowledge-based systems. These, along with reasons for their selection, are described below.

2.1 On-board Diagnostic Systems

An on-board diagnostic/advisory system for some major Space Subsystem for the baseline SS made immediate sense to our group. It can be seen to meet all three high-level evaluation criteria:

1. Is there a problem worth solving? Yes. Crew time is far too scarce and too valuable to be devoted to routine maintenance tasks. In addition, the increased complexity and novelty of Space Station major subsystems over Shuttle subsystems is likely to make manual diagnosis and maintenance nearly impossible without a large increases in the number of ground control personnel.

2. Can knowledge-based systems technology help? Yes. There are dozens of successfully fielded diagnosis and repair advisory systems in commercial and industrial use today operating in ground systems of similar complexity to likely Space Station problems. NASA's own experience with prototype knowledge-based systems is greatest in the diagnostic area.

3. Can the system be built under the timing, cost, and other resource constraints of Space Station? A more difficult question, but again we think the answer is yes. In industry, fielded systems of comparable complexity have cost small numbers of millions of dollars and been fielded in 2-4 years on the average. With the proper evolutionary approach to system construction (initially an advisory mode system with full in-the-loop controls coming later), we see no reason to believe that a properly scoped Space Station system could not be built under comparable constraints.

Four candidate baseline Space Station applications in the diagnostic system category became apparent during our visits to JSC and MSFC. All can be classified as fault detection, isolation, and reconfiguration systems (FDIR):

- **Thermal System**--major pluses were very low technical risk because of slow system dynamics and relative simplicity and the existence of an on-going OAST Demonstration Project already building the prototype for such a system. Major minuses were the relatively low perceived importance and visibility of such a system; neither crew nor ground control personnel saw it as a system that would occupy a great deal of their attention.

- **Communications System**--major pluses were a strong desire on the part of crew for help in understanding the system, the knowledge that communications system failure is not a life-critical issue during otherwise normal Space Station flight, the likelihood that communications system failure could isolate crew from ground support during FDIR, and existence of substantial NASA and contractor ground prototyping of communications systems. A major minus was the somewhat greater technical risk of such a system due to greater complexity than other candidates.

- **Electrical Power System**--major pluses were the NASA- internal power system automation experience at MSFC and the potential lack of crew experience with 20kHz power requiring enormous retraining effort without automation. Major minuses were the technical risk imposed by very fast system dynamics and the lack of heuristic experience with 20kHz systems.

- **ECLSS** (particularly water and air regeneration)--major pluses were lack of crew experience with new Space Station technology and relative technical simplicity of the underlying systems. Major minuses were the life-criticality of such a system and lack of existing internal experience or champions within NASA.

2.1.1 Baseline System I--Communications System FDIR

All four systems are excellent candidates for Baseline automation, and all receive our recommendation for serious consideration. We believe, given enough impetus from Space Station and reasonable expenditures by the work package contractors, all four systems could be built for Baseline. However, for reasons of focus we recognize that Space Station may wish to emphasize one of the those candidate applications. In that case, we strongly favor a Communications System FDIR system. Several factors in particular led us to this decision. First was the strongly favorable attitude of crew and mission controllers for such a system; we believe crew acceptance (and indeed crew enthusiasm) is vital to the success of an Baseline knowledge-based system. A second factor was the importance, but non-life-criticality of most communications systems failures; gains in diagnostic ability will be of great utility to Space Station, but individual failures in diagnosis are unlikely to lead to disastrous actions. Third was the fact that many possible failure modes for the communications system would leave the on-board crew in a position where they would have to perform Communications FDIR without the ability to consult ground-based expertise. Finally, there was extensive work by NASA-internal and work package contractors on prototype systems for Communications FDIR; this experience can provide a solid foundation for the development of a deployable system for Baseline.

2.2 Baseline System II--Ground-Based Crew Time Scheduling

A ground-based system to facilitate and improve upon existing methods for crew-time scheduling is our second recommendation for an Baseline-ready knowledge-based system. We do not wish to restrict this system from becoming on-board in part or whole, but believe that significant utility at Baseline could come from a simpler-to-build ground-based system. This is in contrast to the FDIR system discussed above which needs direct sensor (and later effector) connections to the on-board system in real-time.

We will first describe how this applications meets our high-level evaluation criteria:

1. Is there a problem worth solving? The answer is definitely yes. Scheduling is a ubiquitous activity. An automated scheduling system can support multiple applications: crew scheduling; allocation of scarce resources such as scientific apparatus; allocation of consumables such as power; maintenance scheduling; EVA apparatus and payload operations. In addition, a scheduling system has the capability of being used by multiple types of users ranging from the domain-expert scheduler through operational personnel to the crew itself. The scheduler can also be utilized to explore alternatives through asking "what-if" questions.

The quantitative benefits from even modest improvements in scheduling can be significant. If one could perform more scientific experiments by utilizing precious resources more carefully, or by not having to abandon experiments because of inability to support them (e.g., in the 92nd hour of a 96 hour run), the overall productivity of the Station is enhanced. Also, more effective deployment of the crew can have enormous potential value. If one can save only one hour per day of one crew member (equivalent to 7 and one-half minutes for each of eight crew members per day), then at the estimated $35,000 per hour of non-EVA crew-time-value, the additional 365 hours per year of effective time available amounts to $12,775,000. One might easily argue about the $35,000 figure one way or the other, but whatever reasonable value one might pick, the savings can be staggering. To achieve such savings, dynamic rescheduling capabilities are mandatory.

2. Can knowledge-based systems technology help? Again, the answer is affirmative. There is a critical difference between algorithmic schedulers that have arisen in the field of Operations Research (which are prey to the problems of combinatorial explosion) versus knowledge-based schedulers.

Knowledge-based schedulers can more easily deal with the myriad of constraints which are inherent in complex domains with multiple different resource considerations and implications because of the ability to represent and apply (potentially) conflicting requirements. Knowledge-based schedulers can also respond more easily to changing requirements as constraints change. This latter capability is essential whenever situations change unexpectedly and dynamic rescheduling is required. Such changing circumstances will be part of normal space station life. Adaptive scheduling combined with predictive scheduling is thus required, not just the latter alone. In addition, the calls on resources will be from functional needs which have time-varying priorities (e.g, stopping a given experiment late in its run is much more onerous than if it is stopped early on; consuming precious materials is less desirable early in the 90-day replacement cycle).

An objective of knowledge-based scheduling is to provide for a "satisficing" solution which meets the constraints (or most of them) and reaches the desired goals, rather than use a search-for- optimality approach which may be both less flexible and more time consuming. The exigencies of the space station make it critical to remain both strategically and operationally responsive. Through knowledge-based simulation based on model-based reasoning it is possible to test the output of one or more scheduling runs to quickly evaluate alternatives. Such an approach can yield case solutions which are developed in runs at hundreds of times real time. Such is the situation, for example, in an application being developed by IntelliCorp in conjunction with a semiconductor manufacturer.

There are fundamental differences between scheduling for a seven-day Shuttle mission and a permanent space facility like the Space Station. Without attempting to be exhaustive in this analysis, in the case of the Shuttle or Skylab missions: (a) there are fewer systems for which constraints are to be satisified; (b) there are fewer activities (e.g., numbers of scientific experiments) in progress at any one time, and (c) the crew and ground-based operational personnel and scientific investigators can be more tightly scheduled. With respect to (c), it is one thing to "externally schedule" an astronaut for a seven-day trip and quite another to expect him or her to act in an "automaton mode" for a multiple-month stay. For normal psychological reasons and to benefit from crew-member creativity, we need to provide for more autonomous action.

Instead of scheduling each individual action, it is appropriate to schedule within operational envelopes of opportunity. For example one can schedule overall resources such as crew time, electrical power, thermal-system needs, laboratory instruments (where multiple instruments may be required for a single experiment), computational power, and communications bandwidths within an envelope of time and then allow the crew, ground-operational personnel, and scientific investigators to work on the trade-offs of what should be done when, likely with support from the dynamic-rescheduling mode of the knowledge-based scheduler. Some constraints may only arise within the course of time and may not have been a priori predictable.

3. Can this problem be solved under the time, resource, and organizational constraints of the Space Station? Again, the answer is yes. Knowledge-base scheduling is not a new technology, either outside NASA or within. NASA has already accomplished significant work in the scheduling arena. John Jaap and his group at the Marshall Space Fight Center have developed (beginning in 1979) an Experiment Scheduling Program (ESP) for Spacelab missions. ESP has also been used for scheduling projected 90-day Space-Station mission segments. The system currently is action-based instead of envelope-based, and thus changes are needed to permit more flexible crew-time scheduling. This project has had significant success and should be thoroughly assessed as a starting point for work in support of the Space Station, most likely as a partial model for the next-generation knowledge-based scheduler.

2.3 Baseline System III--Trend Analysis of Network Performance

A system for continually monitoring the on-board computer network for performance improvement and fault detection is our third recommendation for an Baseline-ready knowledge-based system. Such a system meets the three high-level evaluation criteria.

1. Is there a problem worth solving? Yes. The current network design is based on present anticipation of applications that will be carried out on the Space Station in the next two to three decades. It is difficult to completely identify the characteristics of these applications and design a network that can accommodate growing needs of future applications. Further, much hidden information transmitted across the network is not analyzed or recorded (such as recovery from transient errors) because the applications engineers may not realize or anticipate the impact of this information on other applications. The amount of this information is normally very large, and there may not be sufficient processing power or

memory to record it. This information may be useful for predicting performance bottlenecks, potential failure modes, and possible interactions of applications. A system for monitoring network traffic in order to identify possible performance bottlenecks and failures is essential. The objective is to predict failures and performance degradation before they actually occur. The network traffic as well as the content of messages transmitted on the network are useful in making such predictions.

2. Can knowledge-based systems technology help? Yes. A large volume of information is circulated on the network. The processing of this information requires tremendous processing overhead. A knowledge-based approach can be applied to understand the nature of applications, abstract essential information, and pinpoint areas of improvement and potential problems. There are four areas in which a knowledge-based approach would be helpful.

- **Data abstraction and compression**--A large amount of data is being transmitted on the network. This information must be collected and abstracted in real time so analysis can be performed later. Knowledge on the nature of applications, the characteristics of the network and technology, the networking standards, the error-recovery mechanisms, and the message's formats and its points of origination and destination are examples of information that is useful for selecting the appropriate mechanism for data abstraction and compression. New strategies may also have to be devised in real time.

- **Trend analysis**--Information on long-term message traffic and history, such as recovery from transient errors, unpredictable interactions of messages generated from different applications, failure modes related to messages generated before failures, and increased network usage due to interactions of applications, are useful for analyzing future network performance, failures, and growth.

- **Learning new strategies to cope with unforseen situations**--Possible failures and performance degradation can be predicted using a combination of analogy, deduction, and induction methods. Scenarios that have previously occurred may be useful for proposing experiments to verify hypotheses on failures and performance bottlenecks.

- **Advisory system for evolutionary growth and improvement**--A knowledge-based system will be useful for analyzing whether the network capacity will be exceeded, based on the knowledge of possible new applications, and proposing evolutionary growth, based on current traffic patterns.

3. Can this problem be solved under the time, resource, and organizational constraints of the Space Station? The answer is definitely yes. Extensive research has been carried out on knowledge-based performance prediction. A project funded by NASA is being undertaken at the University of Illinois, Urbana-Champaign, to design a knowledge-based system for load balancing on a local computer network. Empirical studies on predicting process resource usage using statistical methods, and network failure and recovery have also been conducted at that site. The system can be implemented at low cost because it does not require real-time processing (except in data abstraction), and information can be analyzed off-line on the ground.

2.4 Baseline System IV--Intelligent Assistance for On-board Reactive Science

We believe that knowledge-based systems can have substantial impact at Baseline in improving the conduct of on-board reactive science by making crew more knowledge and comfortable with complex experiments. It fits the three high-level evaluation criteria as follows:

1. Is there a problem worth solving? Yes. Crew strongly expressed a desire to move from "on-off switch flippers" to knowledgeable postdoctoral-level scientists in conducting a wide variety of experiments on space station. They need to know more, and to learn more as needed, about the goals, mechanisms, and real-time conduct of science under their supervision in order to respond to unexpected and interesting results during the course of long-term experiments. Ground-based PI's of experiments on Spacelab indicated a mutual desire to make crew considerably more informed and capable.

In addition, the changing nature of science aboard Space Station creates problems not seen on previous, shorter duration missions. When experiments only last a few days, it is feasible for a PI to be available at the POCC to evaluate the experiment in real time, monitor data for interesting events, and answer questions throughout the entire experiment. Space Station experiments will have no such built -in time limit, and may, in fact, last many months. It is not reasonable to expect that a PI will be available around the clock for the entire duration of these experiments. Thus, the experiments themselves will need to be more self-monitoring and self-diagnosing. Telemetry limits may also require experiments to be able to filter out uninteresting data from the real-time telemetry.

2. Can knowledge-based systems technology help? Yes, in at least five ways:

- **Data Monitoring**--using knowledge about the experimental goals, devices, and current context to determine if high-quality data is being produced.

- **Rapid Data Analysis**--acting as a full-time laboratory assistant to inform crew when something particularly interesting is happening that might require further attention (interesting might mean expected, but rare results, or unexpected and potentially meaningful new results).

- **Overall Experiment Planning and Scheduling**--keeping track of the original plan for conduct of the experiment, making changes when necessary or replanning when things go wrong (or even when things go unexpectedly right and extra time is available for other tasks).

- **Diagnosis and Repair**--sometimes experimental apparatus breaks down. Relatively small-scale FDIR systems could be built into each major piece of experimental equipment to provide real-time guidance on troubleshooting and repair to crew.

- **Science Augmentation**--understanding what other experimental equipment and resources are available on Space Station to suggest additional devices that can be brought to bear in problem-solving when need or opportunity arises.

3. Can the system be built under the timing, cost, and other resource constraints of Space Station? We believe so. There is a pilot project underway at Ames Research Center and MIT under the guidance of Professor Larry Young, a well-known vestibular physiologist and Spacelab PI. This project uses standard mini-computers and off-the-shelf knowledge-based systems tools to duplicate much of what Professor Young knows about a typical Spacelab Experiment. Since many Space Station experiments are likely to have attached mini-computers of the type Professor Young is using, incremental hardware and software cost is both controllable and separable from other on-board systems. There is also an easy pathway for ground testing and incremental transition to Space Station since little, if any, connection to life-critical Space Station subsystems will be necessary.

2.5 Baseline System V-Space Station Control Center Automation

The use of KBS technology in the Space Station Control Center (SSCC) is an important application that meets all three of the evaluation criteria:

1. Is there a problem worth solving? Yes. A major element of the Space Station life cycle costs will be the expense of maintaining a constant ground support capability. The ground support role in the past has been filled by some of the most experienced and highly trained personnel in NASA. The previous missions were of short duration, however, when compared to the space station. It will be impossible in terms of both cost and number of qualified personnel to staff the station constantly with the same quality personnel that have been used traditionally, for the duration of the Stations operation.

2. Can knowledge-based systems technology help? Yes, in a significant way. The primary role of KBS technology in the SSCC would be to act as an an interface between non-expert personnel and the complex SSCC control consoles. The KBS is thus acting as an electronic experience amplifier. A KBS could be used to capture the operational knowledge from the best support experts, for each of the major subsystems. This knowledge base can be modified easily as operational experience is gained during the actual Space Station design and operation. With the operational experience in electronic form it can be used to off load the heavy staffing requirement from experts to lesser skilled ground controllers.

The KBS can provide information about the actual support task making suggestions as to what is best to do or consider next. It can also offer an on-line record of support decisions complete with justifications for why actions were taken, for later analysis or training. It can provide help facilities not only for the operation of the specific console but also on less used procedures, interfaces to other consoles or operators and diagnostic information. The same KBS that is used in the on line SSCC can also be of benefit in training new SSCC staff off line.

The end benefit is that the SSCC can be staffed with a broader skill mix of personnel, while maintaining the same level of support quality. This mode of operation can free the most skilled personnel for more creative and/or rewarding tasks, e.g., designing/planning system improvements.

3. Can the system be built under the timing, cost, and other resource constraints of Space Station? Yes. The technology needed to build front ends to complex systems is in the "off-the-shelf" rather than R&D category of KBS technology. Many examples of this type of system have been built and are in use today. The average system of this type has been built by one to two people in less than six months for the first prototype.

One of the first well known systems was SACON (see Bennett and Engelmore, SACON: A Knowledge-Based Consultant for Structural Analysis, IJCAI--79, p. 47), built to provide a convenient interface to a complex software structural analysis tool. Since that time, these "intelligent front-ends" have constituted one of the largest and most successful areas for KBS application.

There are other notable examples under development today. One is the Pilot Vehicle Interface (PVI) of the DARPA-supported Pilot's Associate program. The feature of the PA program is that it assumes that the pilot is in control and the KBS is acting as an information manager. Ideally, the PA actually anticipates the pilots "intent" from the operational context of the current situation and acts more as an information manager than a decision maker. The PVI acts as the man-machine interface. Another example is the prototype Integrated Communications Officer (INCO) Expert System Project under development at JSC.

A major distinction between SSCC software and the actual Station software is that this software does not have to be space qualified to the same degree that the on board software does. This means that there can be a broader mix of software packages, there may be a slightly relaxed transition to Ada for some aspects and the supporting console hardware is available commercially today. Many of the aspects that make introduction of KBS technology into the Station difficult are thus removed. This makes the SSCC application one of the least risky while still having a very large impact on overall life-cycle costs.

Because of the relaxed software requirements a significant KBS could be in the SSCC at Baseline. To effectively include KBS capabilities in the Baseline SSCC, however, KBS technology must be factored in from the start. It should not be viewed as an appendage that is added after the "real" SSCC is built, but is tightly integrated into every module and every console. This will require a strong commitment from the SSCC contractor and the responsible NASA managers that can probably only develop as a direct result of a NASA directive of technical preference.

2.6 What We Didn't Choose for Baseline SpaceStation

A few comments are in order about some specific systems we did not choose for Baseline despite wide publicity in the Space Station program or strong suggestion in previous studies or from groups we visited.

- **OMS**--the concept itself, of an overall command and control system for Space Station incorporating the best of knowledge-based systems technology is extremely attractive. Unfortunately, we find the current problem both too nebulous and too difficult for current, off-the-shelf knowledge-based systems technology. Ground-based command and control systems are only beginning to emerge in practical use, and there is no real experience with Space Station command and control upon which to base a heuristic system. We do not mean to preclude pieces of software (such as planning and scheduling systems) necessary for an integrated OMS from appearing on Baseline Space Station. We will describe this as an important area for future growth in the section on advanced development funding.

- **Inventory Management**--a problem suggested by crew (their first choice, in fact), by ground controllers, and by work package managers as a likely target for automation was inventory management of replaceable and renewable resources on Space Station. We think automation of inventory management is a good thing to do, but see little advantage to the use of knowledge-based systems. We know of few, if any, current industrial examples, and believe that traditional OR-based methods combined with reasonable database technology will suffice, at least for Baseline.

- **Training and Instructional Aids**--another important problem and a likely candidate for eventual use of AI on Space Station. However, it does not make our list of suggested Baseline emphases for several reasons. First of all, the technology for true Intelligent Computer Assisted Instruction (ICAI) is still in its infancy and must be considered research rather than off-the-shelf. Second, the instructional AI systems that work well rely on a knowledge base of existing expertise, of which none yet exists for many of the major components of Space Station. Third, the application is not going to be of high visibility for Baseline; it will be very difficult to quantify payoff for Baseline (but much easier for evolutionary life-cycle costs of Space Station). We also include this in our discussion below on advanced development funding.

- **Design Knowledge Capture**--an oft-repeated topic both in Space Station documentation and during our on-site visits during this study. However, we think that Space Station is seriously underestimating the difficulty of achieving "preservation of the corporate memory" in any serious way beyond building enormous databases of design criteria and specifications. We will discuss this problem in some detail below, but believe that serious work should have started ten years ago if it were to significantly help Space Station at Baseline.

- **In-the-Loop Systems**--we believe that the issues of safety, technical risk minimization, and building crew confidence, argue against in-the-loop control systems at Baseline. As stated above, all on-board systems should be designed to eventually and incrementally achieve greater degrees of autonomy, but we think that Baseline should be a time for advisory knowledge-based systems. We will discuss some of the hooks and scars necessary to allow this evolution, below.

3. Recommendations for Hooks and Scars

As stated earlier in this report, much has been written about the broad concepts and classes of hooks and scars and we have no desire to duplicate much of that ground. We do believe that a good way to get to specific requirements on the baseline Space Station that will allow evolutionary growth of knowledge-based systems is to focus on specific baseline choices (such as an FDIR Communications System) and concentrate on evolutionary pathways for that system. The list from our preliminary analysis is as follows:

- Sensors in place wherever potentially useful in the future--certainly at every LRU to monitor I/O behavior.

- Software-controlled switches throughout to allow evolution of Baseline systems from advisory-mode to in-the-loop reconfiguration.

- Adequate Baseline capacity of Standard Data Processors (SDP's) and Embedded Data Processors (EDP's)--some reasonable numbers to work with are 32 bit internal and external (peripheral data) busses, 20mHz (at least 3-4 MIPS machines), and at least 12-16 megabytes of RAM; SUN 3/260 equivalents to pick a popular off-the-shelf example.

- Adequate network capacity--at least 100 megabit/second (which we believe is the current Space Station specification).

- The concept of a database server to provide general database accessibility--in other words, DMS should make it possible for any computer on the Space Station internal network to retrieve data from any other system on the network. This means substantial work on flexible hardware and software interfaces.

- The concept of inherent replaceability with upgraded components--this includes processors, memory, and all other major computational components. We understand that planning for continual addition of new computational devices is a problem when issues like space worthiness are vitally important. However, we point to such major NASA missions as Shuttle and Hubble Space Telescope as examples where inherently unreplaceable computers and related components have had major negative impact on mission success.

- An example of a major computational addition would be the Spaceborne VHSIC Multiprocessor System (SVMS) being developed by OAST. Ideally it should be added simply by plugging into the standard Network Interface Unit (NIU).

- Adequate communications bandwidth between Earth and Space Station.

- The ability for ground-based support personnel (controllers, scientists, etc.) to easily sign on to the Space Station internal computer network on an adequately controlled, need to interact, basis.

- The parallel ability for crew on Space Station to log on to ground-based computer networks to communicate, retrieve information, etc.

- The ability to add and modify software on a routine basis--we feel that Space Station evolution will mean nearly continuous code replacement and addition throughout its lifetime and do not believe the current mechanisms for Shuttle and other large NASA devices are adequate.

- Finally, tolerance for a heterogeneous computational environment on Space Station--this, we realize, is easy to say, but painful to implement. However, we strongly believe that the gain in evolutionary productivity makes it very important.

4. Recommendations for Advanced Development Funding

We believe that the Space Station Level I Program in Evolution and Advanced Development can contribute to the growth of knowledge-based systems in three ways:

1. By making sure several Baseline systems come into existence. This means shepherding the systems we have suggested rapidly through the design and prototyping stage to the point where work package contractors can take over final implementation.

2. By conducting an active prototyping and demonstration program for post-Baseline evolution and new applications. Prior projects have suffered from the lack of a direct link between research and end-users that we believe must exist from even the earliest stage of potential Space Station application. Advanced Development should ensure the availability of appropriate Space Station personnel, get the responsible Level II managers into the loop, and build the bridges to the work package contractors early in the process. Advanced Development should co-fund with OAST and other organizations that do research and build prototypes (DARPA, other DOD agencies, etc.) to make sure Space Station is the domain of choice for many early investigations and to give researchers the added bonus of actual Space Station utilization if their efforts succeed.

3. By strongly encouraging (through OAST or other relevant organizations) research in the following critical areas:

- **Design Knowledge Acquisition**--we believe that three major AI research topics are involved to truly build a computational memory of all the factual, heuristic, and anecdotal information that arises during Space Station design, building, and testing that will be of potential utility during operations. The first topic is automated knowledge acquisition; the current, mainly manual methods of building knowledge bases will not suffice for devices as large and diverse as Space Station. The second topic is combining knowledge from many different sources of information. Current knowledge-based systems may combine expertise from two or three human experts; systems that will be need for serious Space Station use will involve knowledge from many hundreds of experts. The final topic is simply the technology associated with utilization of very large scale knowledge bases. Evolutionary Space Station systems are likely to be three or more orders of magnitude larger than the biggest current knowledge-based systems in commercial use today.

- **Operation Management System**--we recommend making overall command and control of Space Station a topic for serious study and prototyping in the knowledge-based systems world. Two particularly important topics are the interface of knowledge-based systems and operating systems, and the concept of intelligent agents for easy access to wide varieties of complex on-board software systems. In addition, Advanced Development should determine the relevancy of potentially useful military command and control projects and foster technology transfer from those projects to the Space Station domain.

- **Intelligent Computer Aided Instruction (ICAI)**--we believe that Advanced Development should hasten the development of this technology for both on-board and ground-based use for Space Station. In particular, work on the relation of ICAI to intelligent database systems is needed. We would like to see a plan for slow integration of automated training during the evolutionary life of Space Station.

- **Automatic Programming**--software addition and maintenance on Evolutionary Space Station will be an enormous time and cost sink without improvements in this research field. We particularly recommend work on automatic transition from specification to validated code; i.e. software maintainers need only deal with very high-level functional specifications and an automated systems converts specification to code.

- **Knowledge-based Systems Tools in Ada**--although it is perfectly possible to develop systems in one environment and transition them to different operational hardware and software, we believe it is still worthwhile for the SS Advanced Development Program to foster the growth of powerful Ada tools.

Acknowledgments

We would like to gratefully acknowledge the many people that contributed towards the results described in this paper. This group includes, of course, the other members of the Knowledge-Based Systems Study Committee: Jaime Carbonell, David Mishelevich, Brad Allen, Bob Engelmore, and Ben Wah. We would also like to acknowledge the contributions made by the many authors of previous Space Station Automation and Robotics studies, specifically including the reports of the Advanced Technology Advisory Committee. Finally, we would like to thank the many NASA personnel--crew, mission controllers, scientists, engineers, and managers--who generously made available their time for our discussions.

RIA-88-12-05-3

Purposive Discovery Of Mathematical Operator Definitions
Michael Sims and John Bresina — December 1988

In the context of IL, a discovery system for mathematics, we describe our implementation of a general method (called GPP) for the discovery of mathematical operators. This discovery process is driven by the intended purpose of the created operator. The implementation successfully (re)discovered the correct operator definition for multiplication of Conway numbers. The GPP (Generate, Prune and Prove) method is general with respect to the operator's definition language, the specific operators, and the specified purpose of the operator.

RIA-88-12-05-4

Constraint Satisfaction With Delayed Evaluation
Monte Zweben and Megan Eskey — December 1988

This paper describes the design and implementation of a constraint satisfaction system which uses delayed evaluation techniques to provide greater representational power and to avoid unnecessary computation. The architecture used is a uniform model of computation, where each constraint contributes its local information to provide a global solution. We demonstrate the utility of the system by formulating a real-world scheduling problem as a constraint-satisfaction problem..

RIA-89-01-01-03

A Study of Knowledge-Based Systems for the Space Station
Peter Friedland — January 1989

A rapid turnaround study on the potential uses of knowledge-based systems for Space Station Freedom was conducted from October 1987 through January 1988. Participants included both NASA personnel and experienced industrial knowledge engineers. Major results of the study included five recommended systems for the Baseline Configuration of the Space Station, an analysis of sensor hooks and scars, and a proposed plan for evolutionary growth of knowledge-based systems on the Space Station.

RIA-89-01-07-0

A Survey of Computational Learning Theory
Philip Laird — January 1989

This paper presents an overview of formal learning theory from four viewpoints: logic, Bayesian inference, learnability theory, and neural networks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Dates attached

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Titles/Authors - Attached

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Code FIA - Artificial Intelligence Research Branch
Information Sciences Division

8. PERFORMING ORGANIZATION REPORT NUMBER

Attached

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Nasa/Ames Research Center
Moffett Field, CA. 94035-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Available for Public Distribution

Peter Friedland 5/14/92 BRANCH CHIEF

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Abstracts ATTACHED

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT